

Opinnäytetyö (AMK)

Elektroniikka

Elektroniikkasuunnittelu

2015

Tomas Lång

# ARDUINOSTA KAMERAN LANGATON KAUKOLAUKAISIN BLUETOOTH-OHJAUKSELLA



TURUN AMMATTIKORKEAKOULU  
TURKU UNIVERSITY OF APPLIED SCIENCES

Tomas Lång

# ARDUINOSTA KAMERAN LANGATON KAUKOLAUKAISIN BLUETOOTH-OHJAUKSELLA

Uusissa digitaalikameroissa on ominaisuuksia kameran laukaisemiseksi etäältä. Lähes jokaiseen kameraan saa yhdistettyä langallisen laukaisimen, mutta suuressa osassa kameroita on mahdollisuus myös langattomaan etälaukaisuun yleensä wlan-yhteyttä hyödyntäen. Tämä mahdollistaa myös matkapuhelimen käytön ohjaukseen. Vanhoista kameroista, niin analogisista kuin digitaalisistakin, mahdollisuus langattomaan laukaisuun kuitenkin puuttuu. Langallisen laukaisimen porttiin usein saa liitettyä erillisen vastaanottimen, jonka avulla kaukosäätimellä on mahdollista laukaista kamera. Matkapuhelin kulkee kuitenkin lähes poikkeuksetta jokaisen kuvaajankin mukana, joten ylimääräinen laukaisin on ongelma, johon olisi hyvä löytyä ratkaisu.

Tässä opinnäytetyössä rakennettiin kehitysalustan avulla langaton kaukolaukaisin, jota on mahdollista ohjata Bluetooth-yhteyden avulla niin tietokoneella kuin matkapuhelimella. Ohjelmoinnissa käytettiin Arduino IDE:tä sekä Python-kieltä, joiden perusteisiin myös tutustuttiin työn aikana. Työssä rakennettiin käyttöliittymä myös Android-puhelimelle hyödyntäen Googlen lanseeraamaa MIT App Inventor-ohjelmistoa.

Lopputuloksena työssä saatiin rakennetuksi toimiva prototyyppi kaukolaukaisimesta sekä käyttöliittymät tietokoneelle ja puhelimelle. Kytkennän käyttöaika paristoilla testattiin virtamittauksilla ja myös bluetooth-yhteyden kantama mitattiin. Mittausten perusteella lopputulos todettiin speksien mukaiseksi, joten prototypistä olisi helppo kehittää lopullinen tuote. Tulosten pohjalta käytiin myös pohdinta työstä ja sen mahdollisista parannuksista.

## ASIASANAT:

Arduino, kaukolaukaisin, Bluetooth, Python

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Electronics | Electronics Design

2015 |20

Instructor: Timo Tolmunen, D.Sc (Tech)

Tomas Lång

## WIRELESS CAMERA REMOTE TRIGGER WITH BLUETOOTH-CONTROL USING ARDUINO

There are few options to remotely take a photo in new digital cameras. You can connect a wired remote trigger to almost every camera, but many cameras have the opportunity to wireless trigger with wlan connection. That also makes possibility to use mobile phone to control the camera. Older cameras, both analog and digital, there is no option for that. Usually you can connect separate receiver to trigger-connector and control the camera with remote control. Almost every photographer keeps mobile phone with them so separate trigger is a problem, which should find a solution.

In this thesis, a wireless remote trigger was built with a development platform, which can be controlled with computer or mobile phone via Bluetooth-connection. There is Arduino IDE and Python languages used in the programming and the basics of these were reviewed during the work. Also the Android interface was built with MIT App Inventor –program that is launched by Google.

In the final result of thesis was a working prototype of remote trigger and the interfaces to control it with computer and mobile phone. The usage time with battery was tested with current measurements and the distance of Bluetooth was measured. On the basis of measurements the specs was reached, so it would be easy to develop the final product from the prototype. Based on the results, the final work and possible improvements are discussed.

### KEYWORDS:

Arduino, remote trigger, Bluetooth ,Python

# SISÄLTÖ

## KÄYTETYT LYHENTEET

<b>KÄYTETYT LYHENTEET</b>	<b>6</b>
<b>1 JOHDANTO</b>	<b>1</b>
<b>2 KAUKOLAUKAISU</b>	<b>2</b>
2.1 Langalliset laukaisimet	2
2.2 Langattomat laukaisimet	3
2.2.1 Infrapunalaukaisimet	3
2.2.2 Radiotaajuuksilla toimivat laukaisimet	3
2.2.3 WLAN-yhteydellä laukaisu	4
2.3 Kameravalmistajien liittimet	4
<b>3 KEHITYSALUSTAT</b>	<b>5</b>
3.1 Picaxe	5
3.2 LaunchPad	6
3.3 Raspberry Pi	6
3.4 Arduino	7
3.4.1 Arduino Uno R3	7
3.4.2 Arduino Due	8
3.4.3 Arduino Bluetooth	8
<b>4 ARDUINON OHJELMOINTI</b>	<b>9</b>
<b>5 SUUNNITTELU</b>	<b>11</b>
5.1 Arduinon valinta	11
5.2 Bluetooth	12
5.3 Rele	12
5.4 KytKentä	13
5.5 Ohjelmointi	13
5.5.1 Arduino C	14
5.5.2 Python	14
5.5.3 MIT App Inventor	15

<b>6 MITTAUKSET</b>	<b>17</b>
<b>7 POHDINTA</b>	<b>19</b>
<b>8 YHTEENVETO</b>	<b>20</b>
<b>LÄHTEET</b>	<b>21</b>

## **KUVAT**

Kuva 1. Langallinen laukaisin purettuna.	2
Kuva 2. Canon-kameran kaukolaukaisuliitin. [18]	4
Kuva 3. Nikon-kameran kaukolaukaisuliitin. [18]	4
Kuva 4. USART-muunnoksen periaate. [19]	9
Kuva 5. C-kielen ja Arduino-kielen vertailu. [10]	10
Kuva 6. Luvun lukeminen koodista.	14
Kuva 7. MIT App Inventor 2: Ulkoasun suunnittelu.	15
Kuva 8. MIT App Inventor 2: Kommentojen syöttö.	16
Kuva 9. Signaalin voimakkuus laitteen vieressä (vasen) ja 9 metrin päässä (oikea).	18

## **LIITTEET**

Liite 1. Arduino-versioiden vertailu.	
Liite 2. Python-ohjelma.	
Liite 3. Ote Arduino-ohjelmasta.	

## KÄYTETYT LYHENTEET

ARM	32-bittinen mikroprosessoriarkkitehtuuri
DA-konvertteri	Digitaalisanalogimuunnin
I/O	Input Output, sisääntulo ulostulo
PWM-ohjaus	Pulssileveysmodulaatio-ohjaus
USART	Piiri, joka muuttaa rikkakkaisdatan sarjadataksi

# 1 JOHDANTO

Nykyään uusissa kameroissa on mahdollisuus matkapuhelimen yhdistämiseen langattomasti hyödyntämällä kameraan sisäänrakennettua tai lisäosalla toimivaa wlan-yhteyttä. Tämän avulla matkapuhelimella voi myös laukaista kameran ilman erikseen ostettavaa kaukolaukaisinta. Vanhemmissa kameramalleissa tähän ei kuitenkaan ole mahdollisuutta edes lisäosien avulla, vaan ainoa mahdollisuus on erilliset langalliset tai langattomat laukaisimet. Matkapuhelin kulkee kuitenkin lähes poikkeuksetta jokaisen kuvaajankin mukana, joten ongelmaan olisi hyvä löytää ratkaisu.

Tässä opinnäytetyössä perehdytään Arduinoon, ja rakennetaan sen avulla niin matkapuhelimella kuin tietokoneella ohjattava kaukolaukaisin järjestelmäkameraa varten. Työssä tutustutaan Arduinon ohjelmointiin, Python-ohjelmointiin sekä Androidin ohjelmointiin hyödyntämällä Googlen lanseeraamaa MIT App Inventor -ohjelmistoa. Työn lopuksi kytkennän toiminta varmistetaan mittaamalla niin virrankulutus kuin Bluetooth-yhteyden kantama. Lopuksi käydään myös pohdintaa työn parantamiseksi ja prototyypin kehittämiseksi lopputuotteeksi asti.

Internetistä löytyy muutamia artikkeleita aiheesta, kuten esimerkiksi Andreas Lundquistin laatima tutoriaali Evothings.com-sivustolle (löytyy osoitteesta: <https://evothings.com/diy-camera-control-on-your-smartphone-via-an-arduino-uno-ble-shield/>). Kyseinen artikkeli käsittelee kuitenkin vain Canon-merkkisen digitaalikameran etäohjausta usb-väylän kautta ja kyseinen työ vaatii myös kameran ohjelmointia. Tämän lisäksi internetistä löytyy myös artikkeleita langallisille laukaisimille, jotka hyödyntävät Arduinoa älynä ja laukaisijana toimii esimerkiksi liiketunnistin.

## 2 KAUKOLAUKAISU

Kaukolaukaisin on eräänlainen kaukosäädin digitaalikameralle. Sen avulla kuvaajan ei tarvitse olla kameran äärellä, vaan kameran voi laukaista etäältä. Tällä tavalla saadaan esim. vältettyä kameran tärähdys kuvatessa tai otettua kuva, johon kuvaaja itsekin pääsee.

Laukaisimia löytyy markkinoilta paljon erilaisia, niin langallisia kuin langattomia. Langallisen laukaisimen huono puoli on se, että kuvaajan pitää itse olla lähellä kameraa, hyvänä puolena hinta ja toiminta ilman virtalähdettä. Näitä käytetäänkin lähinnä tärähdyksen estämiseksi jalustan kanssa kuvatessa. Langaton laukaisin taas antaa mahdollisuuden siirtyä etäämmälle kamerasta, mutta hinta on yleensä moninkertainen ja laite vaatii virtalähteen, yleensä pariston. [1]

### 2.1 Langalliset laukaisimet

Yksinkertaisimman langallisen laukaisimen toimintaperiaate on hyvin yksinkertainen; laukaisin liitetään kameraan sille tarkoitettuun liittimeen ja laukaisimen sisällä oleva 2-osainen kytkin yhdistää maadoitukseen joko laukaisujohtimen tai tarkennusjohtimen. Virtalähdettä siis ei tarvita lainkaan, koska laukaisin vain kytkee kamerassa jo löytyviä liittimen pinnejä yhteen. Kuvassa (kuva 1) yksinkertainen langallinen kaukolaukaisin purettuna.



Kuva 1. Langallinen laukaisin purettuna.



Langallisia laukaisimia on myös monipuolisempia versioita, jotka sisältävät valotusajan säädön, ajastuskuvauksen ja sarjakuvauksen. Toimintoja ohjataan kaukolaukaisimen näytöltä ja virran laukaisin saa paristosta. Lisätoiminnoista huolimatta toiminta perustuu samaan kuin yksinkertaisimmankin laukaisimen eli laite ohjaa vain laukaisua ja tarkennusta. Tästä johtuen esim. pitkää valotusaikaa ohjelmoitaessa pitää kameran valotusaika säätää ns. bulb-tilaan, jonka avulla kuvaa otetaan niin pitkään, kuin laukaisin on pohjassa. [1]

## 2.2 Langattomat laukaisimet

Langattomaan laukaisuun löytyy useita vaihtoehtoja. Langattomuus on toteutettu wlan-, infrapuna- tai radio-yhteydellä. Laukaisimen valintaan vaikuttaa haluttu etäisyys kameraan ja myöskin kameran mahdolliset integroidut yhteysmahdollisuudet. [1]

### 2.2.1 Infrapunalaukaisimet

Useimmissa järjestelmäkameroissa on infrapunavastaanotin kaukolaukaisinta varten. Infrapunalaukaisun huonoja puolia ovat etäisyys (yleensä noin 5 m) ja suoran näkyvyyden vaatimus, hyvänä puolena edullisuus. Useimmissa soveltuvissa kameroissa vastaanotin on vain etuosassa kameraa, joten laukaisu infrapunalla ei onnistu takaapäin, joka rajaa käyttömahdollisuuksia. Useimmissa infrapunalaukaisimissa ei myöskään toimi tarkennus. [2]

### 2.2.2 Radiotaajuuksilla toimivat laukaisimet

Yleisin toimintatapa langattomissa laukaisimissa on radiotaajuus, joka mahdollistaa jopa 100 m:n kantaman ilman suoraa näkyvyyttä. Vastaanotinta ei kameroissa löydy valmiina, mutta yleensä sekä vastaanottimen että lähetimen voi ostaa yhtenä pakettina. Sama ohjain eli lähetin voi ohjata useampaa vastaanotinta mahdollistaen useamman kameran laukaisun tai esimerkiksi salamavalon ohjauksen. Jokainen vastaanotin ja lähetin vaatii oman virtalähteensä, ja myös hinta on infrapunalaukaisinta kalliimpi. Ominaisuudet ovat vastaavia kuin langallisissakin mallista riippuen. [2]

### 2.2.3 Wlan-yhteydellä laukaisu

Uusimmissa kameroissa on sisäänrakennettuna wlan-yhteys, joka mahdollistaa kuvien langattoman siirron ja kameran etäkäytön. Joihinkin vanhempiin malleihin ominaisuus on saatavilla lisäosan avulla. Kameravalmistajat ovat myös ohjelmoineet mobiilikäyttöä varten etäkäyttöohjelmat. Kameravalmistajasta ja kameramallista riippuen ohjelman avulla voi nähdä etsimen kuvan matkapuhelimesta tai tabletista ja valita tarkennuspisteen sekä laukaista kameran. Joissain tapauksissa myös asetusten säätö on mahdollista. Wlan-yhteyden huonona puolena on virrankulutus niin kamerassa kuin mobiililaitteessakin. [3]

### 2.3 Kameravalmistajien liittimet

Eri valmistajat käyttävät kameroissaan erilaisia liittimiä kaukolaukaisinta varten. Toimintaperiaate on kuitenkin kaikissa sama, joten tämä opinnäytetyö on yhteensopiva jokaiselle kameralle liittintä vaihtamalla. Jos taloudessa on useampaa eri merkkiä käytössä, on helpointa laittaa piiriin tai johtimen päähän pikaliitin, johon helposti saa vaihdettua kameran käyttämän liittimen. Kuvissa kahden yleisimmän merkin, Canon (kuva 2) ja Nikon (kuva 3), liittimet ja pinnijärjestykset. Molemmilla valmistajilla on ollut myös muita liittimiä käytössä, mutta nämä ovat nykyään yleisimmät versiot.



Kuva 2. Canon-kameran kaukolaukaisuliitin. [18]



Kuva 3. Nikon-kameran kaukolaukaisuliitin. [18]

### 3 KEHITYSALUSTAT

Kehitysalustasta käyttäjä voi rakentaa lähes minkä tahansa elektronisen laitteen, esim. valaistuksen ja ilmastoinnin automaattiohjauksen tai vaikka murtohälyttimen. Alusta koostuu sekä fyysisestä piirilevystä, jonka ohjaavana älynä toimii mikrokontrolleri, että tietokoneella hallittavissa olevasta kehitysympäristöstä. Mikrokontrollerin sisäinen suoritin eli mikroprosessori ohjaa sille ohjelmoidulla tavalla sisään- ja ulostuloväylien toimintaa ja näin ollen prosessorin tehokkuus on suorassa yhteydessä koko mikrokontrollerin toiminnan nopeuteen. Tämä onkin yksi seikka, joka on huomioitava kehitysalustaa valittaessa, etenkin kun rakennetaan suurempaa järjestelmää. [4]

Mikrokontrollerivalmistajat ovat kehittäneet omat kehitysalustansa, joista esimerkkinä ovat Picaxe, Texas Instruments LaunchPad sekä Raspberry Pi. Piirilevyn voi myös valmistaa itse komponenteista juottamalla, ja esimerkiksi Arduino tarjoaakin alustojensa kytkentäkaaviot kotisivuillaan. Kehitysalustat eroavat toisistaan ohjelmoitavien liittimien määrässä ja niiden ohjelmoitavuudessa, mikrokontrollerin nopeudessa ja monipuolisuudessa, virrankulutuksessa ja myös ohjelmointikielessä.

Valmiista kokonaisuudesta, jossa mikrokontrolleri on ohjelmoitu itsenäisesti ohjaamaan sisääntuloporttiin saapuvien signaalien avulla ulostuloportteja, käytetään myös nimitystä sulautettu järjestelmä.

#### 3.1 Picaxe

Picaxe on opetuskäyttöön suunnattu kehitysalusta, joka helppokäyttöisyyden vuoksi on levinnyt jo sadoille tuhansille harrastelijoille. Suosioon on vaikuttanut myös sen hinta ja helppo ohjelmoitavuus. Ohjelmointi tapahtuu Picaxe Basic -ohjelmointikielellä tai vaihtoehtoisesti graafisella vuokaaviolla, ja laitevalmistajan ilmainen ohjelmisto sisältää myös simulointimahdollisuuden. [5]

Picaxe-alustasta löytyy useita eri konfiguraatioita 8 – 40 ohjelmoitavalla pinnillä varustettuina. Näistä pinneistä lähes jokainen on ohjelmoitavissa niin uloslähdöksi kuin digitaaliseksi tai analogiseksi sisääntuloksi. Alustassa käytetään standardisoitua PIC-mikropiiriä, johon on valmiiksi esiohjelmoitu USB-kaapelilla ohjelmoinnin mahdollistava ohjelmisto ja sama kaapeli sopii jokaiseen Picaxe-alustaan. [5]

### 3.2 LaunchPad

LaunchPad on Texas Instrumentsin kehittämä kehitysalusta, jonka suurin etu muihin kehitysalustoihin on sen virrankulutus, joka parhaimmillaan valmiustilassa on alle 1  $\mu$ A. Myös hinta on yksi vahvuus, sillä halvimmillaan LaunchPad maksaa ti.com-sivustolla 9,99 \$ eli noin 9 €. Texas Instruments valmistaa kehitysalustoihinsa myös lisäosia, joita kutsutaan nimellä BoosterPack. Näiden lisäosien avulla kehitysalustoja voidaan laajentaa langattomilla yhteyksillä, näytöillä, liitäntöillä ja monilla muilla tavoilla. [6, 7]

Ohjelmointiin TI tarjoaa Energia IDE ja Code Composer Studio –ohjelmistoja. Energia IDE on processing-kieleen perustuva ohjelmisto, ja siinä toimiikin useimmat samat komennot kuin Arduinossa. Myös C-kieli on tuettu, ja ohjelmisto perustuu täysin avoimeen lähdekoodiin. Ohjelma ei kuitenkaan tue kaikkia kehitysalustoja, joten vaihtoehtona on Eclipse-pohjainen Code Composer Studio, joka tukee myös kaikkia TI:n prosessoreita, joten sillä on mahdollista ohjelmoida niitä myös erillään. [6]

### 3.3 Raspberry Pi

Raspberry Pi poikkeaa muista alustoista, sillä kyseessä on luottokortin kokoinen tietokone, johon voi suoraan liittää näytön ja näppäimistön. Käyttöjärjestelmä ladataan muistikortille, ja sen avulla laitetta voikin käyttää normaalina tietokoneena. Yleensä käyttöjärjestelmänä toimii Linux, mutta vaihtoehtona esim. Android on yleisesti käytetty. Prosessorina uusimmassa Raspberry Pi Model B:n 2. versiossa käytössä on ARM Cortex-A7 -neliydinprosessori 900 MHz:n kellotaajuudella, jota käytetään myös älypuhelimissa. [7]

Raspberry Pi:ssä olevat GPIO-liitäntänaamat ovat ainoastaan digitaalisia, mikä rajoittaa laitteen käyttöä kehitysalustana. Useimmiten liitäntöjä käytetäänkin esim. Arduinon tai muun kehitysalustan liittämiseksi Raspberryyn, ja näin saadaankin helposti rakennettua yksinkertainen ja vähän virtaa kuluttava käyttöjärjestelmän ja mittauslaitteiston sisältävä kokonaisuus. [8]

### 3.4 Arduino

Arduino on yksi maailman tunnetuimmista avoimeen lähdekoodin perustuvista kehitysalustoista. Avoin lähdekoodi mahdollistaa sen kytkentäkaavioiden saatavuuden ja muokattavuuden kaikille käyttäjille. Arduinon ideaan on alusta asti kuulunut myös koodien ja ohjeiden esteetön jakaminen, ja niitä löytyykin Arduinon kotisivuilta ja monilta muilta alan internet-sivustoilta. [13]

Ohjelmointialusta koostuu mikrokontrollerista, usb-liitimestä ja lukuisista liitännäportteista, joihin voidaan liittää erilaisia elektroniikan komponentteja, antureita, moottoreita, led-diodeja jne. Jokainen pinni on ohjelmoitavissa joko tulo- tai lähtöportiksi, ja niiden joukosta löytyy myös sarjaliikennettä varten omat liitännät. Osa liitännöistä on analogisia, joita käytetään mm. erilaisten antureiden arvojen lukemiseen jännitearvoja lukemalla. [9]

Arduino-kehitysalustoja löytyy useita eri versioita, joista jokaisesta löytyy Atmelin valmistaja mikroprosessori. Versiot eroavat fyysisten mittojen lisäksi toisistaan liitännämahdollisuuksien, käyttöjännitteiden ja muistimäärien osalta. Tässä osiossa käydään läpi kaksi tunnetuinta, Uno ja Due, sekä opinnäytetyöhönkin liittyvä Arduino Bluetooth. Lisäksi liitteestä 1 löytyy laajempi vertailutaulukko Arduinon eri versioista.

#### 3.4.1 Arduino Uno R3

Arduino Uno R3 on USB-väylää käyttävistä kehitysalustoista uusin, ja sitä pidetään muiden Arduino-alustojen referenssinä. Alusta perustuu Atmel ATmega328 -mikrokontrolleriin ja sen käyttöjännite on 5 V. Alustasta löytyy 6 analogista tuloa ja 14 digitaalista I/O-pinniä, joista 6 on PWM-ohjattuja.

Uno R3 käyttää muista alustoista poiketen ATmega16U2-mikroprosessoria USB-sarjaliikenne-konvertterina, kun muissa tähän käytetään FTDI:n valmistamaa FT232RL-piiriä. Piirin vaihdoksen myötä Mac-käyttäjien ei tarvitse enää asentaa ajureita tietokoneelle ohjelmointia varten. Käyttäjät voivat myös ohjelmoida uuden piirin avulla laitteen käyttäytymään tietokoneen lisälaitteena, esim. hiirenä tai levyasemana. Piiri on myös halvempi, joten myös Arduinon hinta on saatu edullisemmaksi. [12]

### 3.4.2 Arduino Due

Arduino Due on ensimmäinen Arduino-kehitysalusta, joka käyttää ARM-pohjaista mikrokontrolleria, Atmel AT91SAM3X8E. Prosessorista johtuen käyttöjännite on 3,3 V, joka on otettava huomioon I/O-pinnejä käytettäessä. I/O-pinnejä alustasta löytyy 54 kpl, joista 12 on PWM-ohjattuja. Lisäksi alustassa on 12 analogista tuloa ja myös 2 analogista lähtöä, jotka on toteutettu DA-konvertterin avulla. [13]

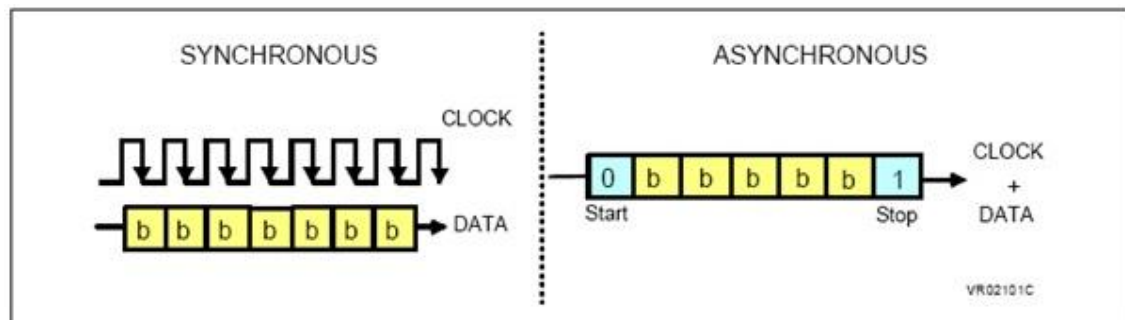
### 3.4.3 Arduino Bluetooth

Arduino BT:n uusin versio eroaa Bluetoothin lisäksi Uno R3:sta usb-liittimen puutteella ja mahdollisuudella käyttää pienempää sisääntulojännitettä. Piirissä käytetty mikrokontrolleri on myös Uno R3:sta poiketen pintaliitosversio Atmega328:sta, joka tietysti hankaloittaa sen vaihtoa rikkoutuessaan.

Arduino BT:ssa käytetään Bluegiga WT11 –bluetoothpiiriä, jonka siirtonopeus on 115 200 b/s ja se on yhdistetty alustan RX- ja TX-pinneihin. [13]

## 4 ARDUINON OHJELMOINTI

Arduinon ohjelmointi poikkeaa ATmega-prosessorien normaalista ohjelmoinnista siten, että siihen ei tarvita erikseen ICSP-ohjelmointilaitetta. Arduinoihin asennettuihin mikrokontrollereihin on valmiiksi asennettu ohjelma, joka mahdollistaa sarja-USART-yhteyden ohjelmointia varten. Tämän lisäksi Arduinoissa löytyy myös toinen mikroprosessori, joka toimii USB-USART-muuntimena, lukuunottamatta Arduino Leonardoa, jossa muunnin on integroituna piirin yhteen ja ainoaan mikrokontrolleriin. Kuvassa (kuva 4) USART-muunnoksen periaate. Ohjelma voidaan ladata suoraan USB-portista Arduinoon, joka näiden muuntimien avulla tallentaa ohjelman mikrokontrollerin luku- eli ROM-muistiin, jossa tieto säilyy myös virran katkettua. [12]



Kuva 4. USART-muunnoksen periaate. [19]

Arduinon kehitysympäristö eli Arduino IDE on tietokoneelle ladattava ohjelmisto, jolla luodaan alustan tehtävät määrittelevä ohjelma eli luonnos. Kehitysympäristön käyttämä kieli perustuu Processing-kieleen, jonka käskyt ovat peräisin C-ohjelmointikielestä. Ohjelmoinnin voikin suorittaa täysin myös C-kielellä, mutta ohjelmista tulee näin huomattavasti pidempiä ja vaikeampia tulkita. Kuvassa (kuva 5) on yksinkertaisesta led-diodin vilkutuksesta esimerkkikoodi sekä C-kielellä (ylempi) että Arduino-kielellä (alempi). Yksinkertaisemman koodin mahdollistaa Arduino-tiimin tekemät valmiit ohjelmakirjastot, joista peruskomennot ovat valmiina ohjelmituna Arduino-kieleen. Kirjastoja löytyy lisää lukuisiin eri käyttötarkoituksiin, esim. moottorin tai näytön ohjaukseen, Arduinon Internet-sivustolta [11].

*Listing 1-1. Blink LED with avr-libc*

```
#include <avr/io.h>
#include <util/delay.h>

int main(void) {
    while (1) {
        PORTB = 0x20;
        _delay_ms(1000);
        PORTB = 0x00;
        _delay_ms(1000);
    }
    return 1;
}
```

*Listing 1-2. Blink LED with Arduino*

```
void setup() {
    pinMode(13, OUTPUT);
}

void loop() {
    digitalWrite(13, HIGH);
    delay(1000);
    digitalWrite(13, LOW);
    delay(1000);
}
```

Kuva 5. C-kielen ja Arduino-kielen vertailu. [10]

Arduinon ohjelmoinnissa tarvitaan aina vähintään kaksi osiota; `setup()` ja `loop()`. `Setup()`-funktiossa määritellään esimerkiksi ohjelmassa käytettävät pinnit ja mahdolliset sarjaliikenneasetukset. Funktiota kutsutaan ohjelman aikana vain käynnistyttyä yhteydessä eli asetukset jäävät voimaan pysyvästi. `Loop()`-funktio sisältää taas itse ohjelmakoodin, jota Arduino lähtee suorittamaan nimensä mukaisesti toistuvasti käynnistyttyään. Funktioiden on aina oltava samassa järjestyksessä, ensin `setup()` ja sitten `loop()`, koska Arduino-kielessä ohjelmat luetaan aina järjestyksessä ylhäältä alas. [10]



## 5 SUUNNITTELU

Tavoitteena työssä on tehdä sekä tietokoneella että Android-puhelimella ohjattava langaton kaukolaukaisin järjestelmäkameralle. Ohjaavan laitteen ohjelmasta tulee löytyä näppäimet tarkennukselle, laukaisulle ja ohjelman lopetukselle sekä myös säätömahdollisuus kameran valotusajalle.

Suunnittelussa on otettava huomioon laitteen kannettavuus. Toisin sanoen laitteen on oltava joko akku- tai paristokäyttöinen, ja sen myötä myös virrankulutuksen on oltava mahdollisimman alhainen. Kuvausolosuhteet voivat myös olla vaihtelevia, joten myös käyttölämpötilat tulee ottaa huomioon. Tässä työssä ei kuitenkaan lähdetä toteuttamaan kotelointia, joten IP-luokitukseen liittyviä määrittelyjä ei oteta huomioon.

### 5.1 Arduinon valinta

Työn voisi aloittaa komponenttitasolta rakentamalla ympäristöä mikroprosessorin ympärille tai käyttämällä valmiiksi markkinoilla olevia kaukolaukaisimia pohjana. Valmiissa kaukolaukaisimissa ongelmaksi koituu piirikaavioiden puute, koska valmistajat eivät niitä julkaise kaikille vapaaseen käyttöön. Komponenttitasolta aloittaessa toki saisi rakennettua hyvinkin kompaktin ympäristön, jossa ei olisi mitään ylimääräistä, mutta valmiilla kehitysalustalla päästään samaan lopputulokseen toimintojen kannalta, ja hinnassakaan ei nykyään ole juurikaan eroa.

Arduinollekin toki löytyy vaihtoehtoja, mm. Raspberry Pi, joka kuitenkin eroaa Arduinosta monessakin asiassa. Yksi suurista eroista on se, että Arduino tarvitsee vain kevyen ohjelmakoodin antamaan käskyjä, kun taas Raspberry Pi vaatii käyttöjärjestelmän toimiakseen. Lisäksi Raspberry Piä löytyy paljon ylimääräisiä liitäntöjä, joita tässä työssä ei tarvita ja hintakin on huomattavasti kalliimpi.

Suurimmalla osalla mikroprosessorivalmistajista löytyy myös valmiita kehitysalustoja, kuten esimerkiksi Texas Instrumentsin TI Launchpadit, jotka ovat jopa halvempia kuin Arduinot. Arduinoa puoltaa sen helppo ohjelmoitavuus, tarvikkeiden ja lisäosien saatavuus sekä avoimen lähdekoodin ja yhteisöjen tuomat edut.

Työhön valikoitui jo olemassa oleva Arduino Uno R3, joka kaikin puolin soveltuu prototyyppin rakentamiseen. Käyttölämpötila saa vaihdella välillä  $-40\text{ }^{\circ}\text{C}$  ja  $+85\text{ }^{\circ}\text{C}$ , joka soveltuu hyvin vaihteleviin olosuhteisiin. [16]

## 5.2 Bluetooth

Vaihtoehtoja langattomalle yhteydelle rajoittaa matkapuhelimesta löytyvät vaihtoehdot, jotka ovat Bluetooth ja wlan. Näistä Bluetooth-verkko on ominaisuuksiltaan riittävä ja helpommin ohjelmoitavissa. Toisaalta myös Arduinosta löytyy versio sisäänrakennetulla Bluetooth-moduulilla.

Moduuliksi työhön valikoitui edullinen JY-MCU-moduuli sen pienen koon ja riittävän, noin 9 m:n kantaman vuoksi. Kokoa laitteella on vain  $4,4\text{ cm} \times 1,6\text{ cm} \times 0,7\text{ cm}$  ja hinta verkkokaupasta tilattuna 6,62 € toimituksineen. Bluetooth-piirinä käytössä on CSR BC417, jonka käyttölämpötila-alue on  $-40\text{ }^{\circ}\text{C} - +150\text{ }^{\circ}\text{C}$ , joten se sopii käyttötarkoitukseen. [14]

Moduulin kytkentä on hyvin yksinkertainen, koska se ei tarvitse kuin käyttöjännitteen ja sarjaliikenneportin. Kytkennöissä kuitenkin on huomioitava Arduinon ja moduulin välinen jännite-ero sarjaliikenteessä. Arduino käyttää TTL-tasoista yhteyttä, jonka huippujännite on 5 V, kun taas moduulin huippujännite on 3,3 V. Tämä on kuitenkin korjattavissa hyvin helposti kahden vastuksen jännitejaon avulla.

## 5.3 Rele

Kaukolaukaisimessa yksinkertaisesti kytketään kameran liittimestä löytyviä tarkennus- ja laukaisupinnejä maadoituspinniin. Langallisilla laukaisimilla tämä hoituu helposti perinteisillä kytkimillä. Kytkimen ja releen toimintaperiaate on sama, mutta rele on sähköisesti ohjattavissa, joten se sopii työhön mainiosti.

Työhön valittiin Keyes 2R1B –relekortti, jossa 2 relettä voidaan ohjata Arduinon digitaalisista porteista saatavalla TTL-signaalilla. Hintaa releelle tuli kaikkiaan 3,12 €. Releen käyttölämpötila-alue  $-25\text{ }^{\circ}\text{C} - +70\text{ }^{\circ}\text{C}$  riittää hyvin tähän projektiin. [15]

## 5.4 KytKentä

Kaukolaukaisin tulee käytettäväksi Nikon-merkkiseen järjestelmäkameraan, jossa on käytössä laukaisinta varten Mc-dc2-nimellä oleva portti. Porttiin sopivaa liitintä ei erikseen myydä, joten työssä käytetään hyödyksi olemassa olevan vanhan langallisen kaukolaukaisimen johtoa ja liitintä, jossa löytyy johtimet maadoitusta, laukaisua ja tarkennusta varten. Nämä yhdistetään releihin siten, että toinen rele kytkee maihin tarkennusjohtimen ja toinen laukaisujohtimen. Jotta kameraa ei tarvita ohjelmointivaiheessa, liitetään releihin led-diodit kytkeytymään vastuksen kautta käyttöjännitteeseen, joiden avulla nähdään helposti kytkennän toiminta.

Relekortin jännite saadaan suoraan Arduinon käyttöjänniteliittimistä. Ohjauspinnin *IN1* ja *IN2* voidaan kytkeä mihin tahansa Arduinosta löytyvistä digitaalisista I/O-porteista, jotka tässä työssä valitaan 7- ja 8-porteiksi.

Bluetooth-moduulin käyttöjännite kytketään samalla tavalla kuin releenkin. Moduulin sarjaliikennettä Arduinolle lähettävä *TXD*-nasta voidaan kytkeä suoraan Arduinon vastaanottavaan *RX*-nastaan eli digitaalisten I/O-porttien 0-nastaan. Aikaisemmin ilmi käynyt jännite-ero on otettava huomioon moduulin vastaanottavaa nastaa kytkettäessä. *RXD*-nasta kytketään siis jännitejakajan avulla Arduinon *TX*-nastaan eli 1-nastaan.

Ohjelmointivaiheessa käyttöjännite saadaan suoraan usb-liittimen kautta tietokoneesta. Jotta laitetta voidaan käyttää myös ilman tietokonetta, kytketään 9 V:n paristo Arduinon *GND*- ja *Vin*-pinneihin.

## 5.5 Ohjelmointi

Ohjelmoinnissa käytetään Arduinoa varten Arduino C –kieltä ja Windows-ympäristöä varten Python-ohjelmointikieltä. Android-järjestelmän voisi ohjelmoida samalla Python-ohjelmalla pienin lisäyksin, mutta tässä työssä tutustutaan sen sijasta MIT App Inventor –ohjelmistoon.

Tässä luvussa käydään ainoastaan tärkeimmät yksityiskohdat. Laajemmat ohjelmakoodit löytyy liitteistä 2 ja 3.

### 5.5.1 Arduino C

`Setup()`-funktiossa määritellään käytettävät I/O-pinnit lähtöporteiksi sekä avataan sarjaliikenne komennolla `Serial.begin(9600)`, jossa sulkeissa oleva arvo kertoo Arduinolle moduulin vaatiman siirtonopeuden. `Loop()`-funktiossa etsitään saapuvaa sarjaliikennettä komennolla `if(Serial.available())`. Jos sarjaliikennettä löytyy, hypätään aliohjelmaan, jossa luetaan saapuvaa dataa sarjaportista niin kauan, kunnes datan joukosta löytyy vaadittu merkki, joka antaa luvan edetä ohjelmassa eteenpäin.

Valotusaikaa lukiessa sarjaliikenteen joukosta on huomioitava se, että Arduino lukee sarjaliikennettä merkki kerrallaan. Tästä syystä tarvitaan käskyjono, jonka avulla saadaan merkeistä luotua kokonainen luku (kuva 6).

```
bitti = Serial.read();
Serial.println(bitti);
if (bitti > 0 && bitti != '/') {
    sarjadata = sarjadata * 10 + bitti - '0';
}
}
```

Kuva 6. Luvun lukeminen koodista.

Käskyn avulla saadaan esimerkiksi merkkijonosta "1,2,3,/" muodostettua kokonaisluku 123, kun ohjelma antaa merkin "/" luvun 3 jälkeen. Tässä tapauksessa "/"-merkki toimii siis luvun katkaisijana.

Arduino C –ohjelma käydään läpi liitteessä 3.

### 5.5.2 Python

Python-kielessä saadaan itse komentojen lisäksi myös rakennettua käyttöliittymä, ja tältä osin se eroaakin suuresti Arduino C:stä. Python-ohjelmassa tässä työssä Arduinon `loop()`-funktiota vastaakin itse käyttöliittymän ylläpito. Käyttöliittymässä jokaiselle näppäimelle annetaan komento, joka määrittelee seuraavan funktion. Myöskään sarjayhteyttä ei pidetä jatkuvasti avoinna, vaan avaavaa komentoa kutsutaan aina funktioissa erikseen.

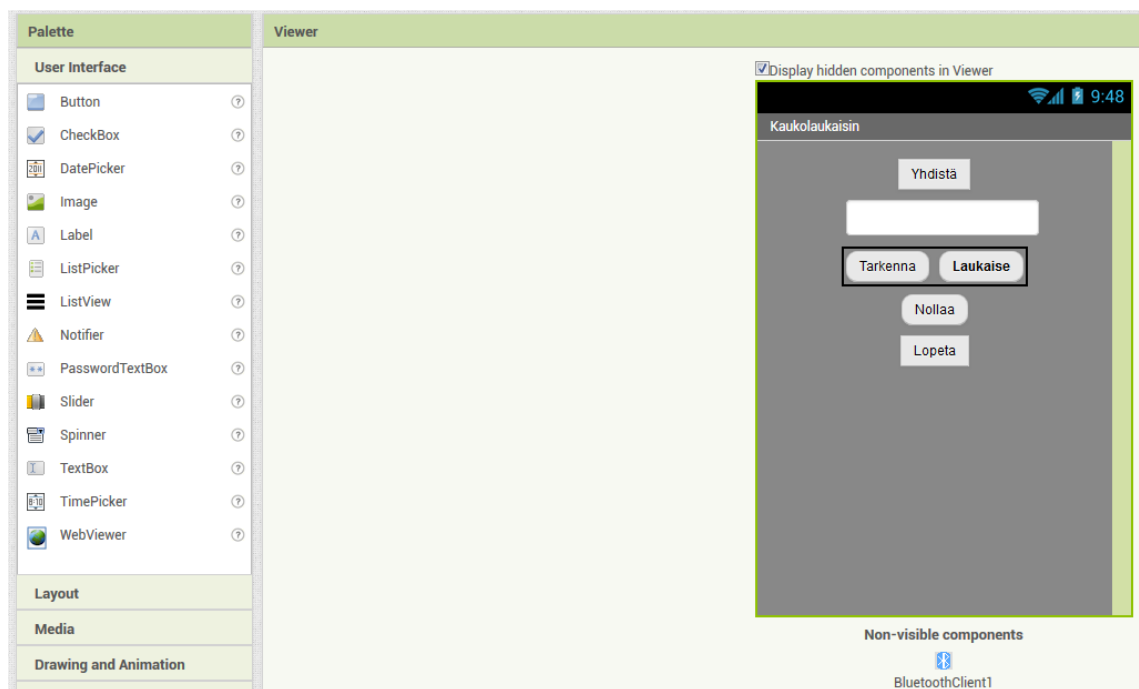
Ohjelmassa käytetään kolmea erikoismerkkiä käskyjen antamiseen. Näistä "/"-merkki annetaan valotusarvon loppuun automaattisesti, jotta Arduino osaa lukea oikean luvun.

Kun arvo on annettu ja laukaisunäppäintä painetaan, lähettää ohjelma merkin "#", joka Arduinon ohjelmassa on merkitty antamaan laukaisukomento. Lisäksi vielä merkki "%" antaa käskyn lopettaa ohjelma.

Python-ohjelma käydään läpi paremmin liitteessä 2.

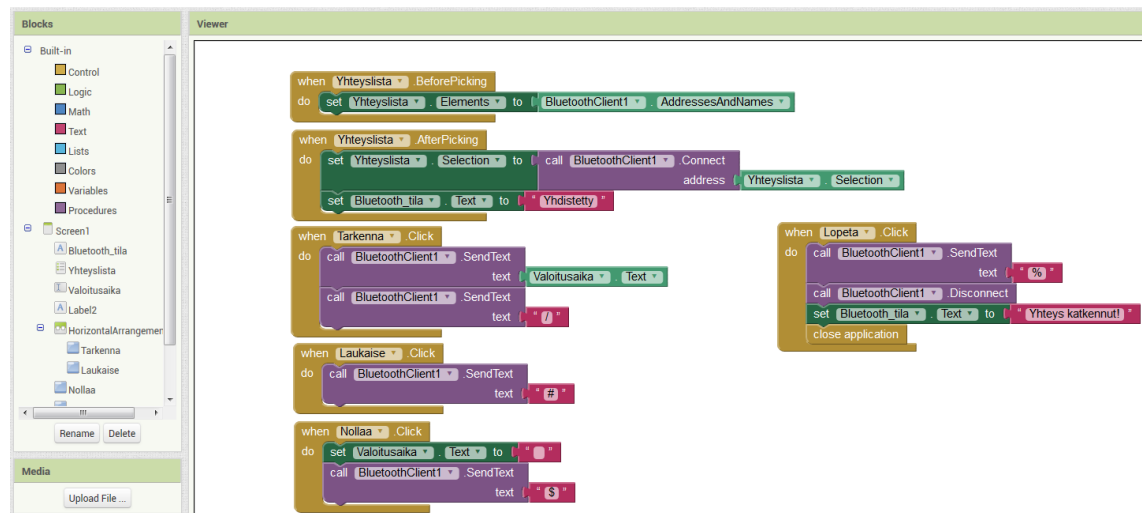
### 5.5.3 MIT App Inventor

MIT App Inventor on Googlen lanseeraama, selainpohjainen graafinen ohjelmointityökalu Android-käyttöjärjestelmää varten. Siinä tekstipohjaisen ohjelmoinnin sijaan ohjelmaan rakennetaan ensin ulkoasu vetämällä valikosta käyttöliittymän osia, esim. tekstikenttiä ja näppäimiä, selaimella näkyvään puhelinsimulaattoriin (kuva 7).



Kuva 7. MIT App Inventor 2: Ulkoasun suunnittelu.

Kun ulkoasu on saatu valmiiksi, siirrytään toiseen välilehteen, jossa annetaan ohjelmassa oleville toiminnoille komennot, jälleen vastaavasti vetämällä valikosta ruudulle ensin haluttu toiminto ja siihen yhdistämällä toiminnon jälkeen tapahtuva komento (kuva 8).



Kuva 8. MIT App Inventor 2: Komentojen syöttö.

Tässäkin ohjelmassa käytetään samoja erikoismerkkejä kuin aiemmin, joten ohjelma on täysin yhteensopiva aiemmin luodun Arduino-ohjelman kanssa.

Android-puhelimen Play-kaupasta on ladattavissa MIT AI2 Companion –ohjelma, jonka avulla luodun ohjelman näkee reaaliajassa puhelimen näytöltä puhelimen ollessa yhdistettynä internetiin. Ohjelmaa pystyy myös kokeilemaan samalla ja näin ohjelmaa on erittäin helppo muokata ja korjata. Lopuksi ohjelman voi tallentaa ja kopioida esim. muistikortille puhelimeen siirtoa varten, jonka jälkeen ohjelma toimii itsenäisesti ilman Internet-yhteyttä.

## 6 MITTAUKSET

Työn mittauksissa selvitetään laitteiston pariston kestoisuutta sekä bluetooth-yhteyden kantamaa. Paristona käytössä on 9 V:n Varta 6LP3146 ja mittalaitteena Meterman 30XR -yleismittari. Lisäksi Bluetooth-yhteyden mittaamiseen käytetään Android-puhelimen Play-kaupasta ladattavaa Bluetooth Signal –ohjelmistoa.

Pariston kestoa tarkasteltiin niin jännitteen kuin virran näkökulmasta. Virtaa mitatessa ongelmaksi koitui virran suuruuden vaihtelu, joka osittain johtuu Bluetooth-yhteydestä ja sen signaalin vaihtelevuudesta. Alla olevassa taulukossa (taulukko 1) näkyy mittaustulokset virrankulutukselle ja sen avulla lasketulle kuormalle eri käyttövaiheissa.

Taulukko 1. Virrankulutus ja kuorma eri käyttövaiheissa.

Toiminto	Virrankulutus (min) [mA]	Virrankulutus (max) [mA]	Virrankulutus (ka.) [mA]	Laskettu kuorma [ $\Omega$ ]
Hakee BT-yhteyttä	60,2	99,6	79,9	109
Yhdistetty ("lepotila")	54,0	55,2	54,6	161
Tarkennettu	135,1	152,6	143,9	61
Laukaistu	219,4	240,6	230,0	38

Pariston datalehdessä (liite 2) löytyy esimerkki, jossa 180  $\Omega$  kuormalla pariston jännitteen alentumiseen arvoon 5,4 V kestää 753 min ja arvoon 4,8 V 840 min. Mittaustulosten mukaisesti ns. lepotilan kuorma on noin 160  $\Omega$  ja piirin vaatima käyttöjännite on vähintään 5 V, joten pariston kestoksi voidaan laskea noin 800 min eli 13 h 18 min.

Kuten taulukosta huomaa, virrankulutus kasvaa huomattavasti sekä tarkennus- että kuvanottohetkellä. Tarkennusvaiheessa virrankulutus lähes kolminkertaistuu, ja kuvaa ottaessa nelinkertaistuu lepotilaan nähden. Jokainen otettu kuva siis hetkellisesti nelinkertaistaa virrankulutuksen, joten karkeasti voidaan ajatella pariston kestävän noin 200 min, jos kuvia otettaisiin taukoamatta. Tässäkin kuitenkin on otettava huomioon jokaisen kuvanoton hetkelliset virtapiikit, joten todellisuudessa kesto on vielä pienempi.

Virtamittausten jälkeen mitattiin myös jännitettä. Tässä vaiheessa laitteiston annettiin olla tunnin ajan lepotilassa, ja jännite mitattiin sekä testin alussa että lopussa. Tunnin aikana jännite aleni arvosta 8,31 V arvoon 7,09 V. Tämä saattaa tuntua aluksi hieman oudolta aikaisempaan mittaustulokseen verrattuna, mutta datalehdessä löytyvän kuvaajan mukaan noin 8 V:n kohdalla jännitteen laskeminen nopeutuu huomattavasti. Tämän vuoksi käytettyä paristoa käyttäessä myös pariston ikä on suhteessa huomattavasti pienempi kuin uutta paristoa käyttäessä.

Ilmoitettu toimintasäde Bluetooth-moduulille piti hyvin paikkansa. Mittauksissa käytettiin apuna Bluetooth Signal-ohjelmistoa, jonka perusteella signaalin voimakkuus noin 10 cm:n päässä oli -42 dBm ja noin 9 m:n päässä -77 dBm. Kantama on käyttötarkoitukseen todella hyvin riittävä. Kuvassa (kuva 9) näkyy kuvakaappaus ohjelmasta kahdella eri etäisyydellä.



Kuva 9. Signaalin voimakkuus laitteen vieressä (vasen) ja 9 m:n päässä (oikea).



## 7 POHDINTA

Työ aloitettiin hyvin yksinkertaisesti testaamalla Arduinon toiminta kytkemällä yhteen digitaalisista liitäntänaistoista led-diodi ja ohjelmoimalla Arduino kytkemään kyseinen liitäntä päälle ja pois säännöllisesti. Kun Arduinon toiminta oli varmistettu, rakennettiin kytkentä kahden releen ohjausta varten ja varmistettiin vastaavalla tavalla myös releen toiminta. Kytkentään ei tämän jälkeen tarvinnut lisätä kuin Bluetooth-moduli ja alkaa pohtia lopullista ohjelmaa.

KytKentä siis oli todella yksinkertainen, mutta heikon ohjelmointitaustan vuoksi itse ohjelman tekeminen oli haastavin osuus ja tämän vuoksi kyseinen aihe valikoitui riittävän haastavaksi opinnäytetyöhön. Ohjelmaa tehdessä tuli vastaan useampia ongelmia: välillä yhteyttä laitteiden välille ei saatu syntymään, välillä kytkentä jäi ikuisen luuppiin, ja eipä yhteys tietokoneen ja Arduinon välilläkään aina ihan moitteettomasti toiminut. Ongelmista kuitenkin selvittiin, ja yksi suuri helpotus tähän oli Arduinon laaja verkostuneisuus, joten apua löytyi monilta niin suomalaisilta kuin ulkomaisilta sivuilta. Työn aikana tuli opittua paljon Arduinon ohjelmoinnista ja myös Python-ohjelmointikieli avautui hyvin.

Työn jatkoa voisi olla mahdollisen koteloinnin suunnittelu sekä myöskin kytkennän fyysisen koon pienentäminen. Tähän pystyy jo vaikuttamaan helposti käyttämällä Arduinon Bluetooth-yhteyden sisältävää alustaa, mutta releen koko jää siinäkin isoksi. Myöhemmin voisi selvittää releen mahdollisen korvaamisen jollain muulla kytkentätavalla. Myös kytkennän integroiminen kameraan olisi looginen jatko työlle ja siinä voisi ajatella ns. pystykuvauskahvan hyödyntämistä laitteen kotelona. Tämän avulla voisi olla mahdollista saada myös virransyöttö suoraan kameran akusta. Ohjelmistoa taas voisi kehittää esimerkiksi sarjakuvauksen mahdollistamiseksi tai kellonajan avulla ohjelmoitavaksi.

## 8 YHTEENVETO

Opinnäytetyössä suunniteltiin ja rakennettiin prototyyppi langattomasta kaukolaukaisimesta Arduinon avulla. Kaukolaukaisinta ohjataan joko tietokoneella tai Android-puhelimella, joihin molempiin ohjelmoitiin oma käyttöliittymänsä. Käyttöliittymän avulla pystyy laukaisun lisäksi säätämään valotusaikaa, edellyttäen kameran valotusajan olevan säädettyä ns. bulb-tilaan.

Kytkenän toiminta varmistettiin mittauksin ja testikäytöllä. Myös paristojen kesto tutkittiin, ja kytkennällä päästiin hieman yli puolen vuorokauden valmiusaikaan Bluetooth-yhteyden ollessa aktiivinen sekä lähes 200 minuuttiin jatkuvalla kuvauksella. Bluetooth-yhteyden kantamaksi saatiin lähes 10 m.

Työssä tutustuttiin sekä Arduinon ohjelmointiin että Python-ohjelmointiin. Androidin käyttöliittymän rakentamiseen käytettiin MIT App Inventor –ohjelmistoa. Työssä käytiin myös läpi kehitysalustojen ja Arduinon perusteita ja eri versioiden eroavaisuuksia.

## LÄHTEET

- [1] When To Use A Cable Release and Remote Shutter Release. [www-dokumentti].  
Saatavilla: <http://www.digital-photo-secrets.com/tip/3405/when-to-use-a-cable-release-and-remote-shutter-release/> (Luettu 25.11.2015)
- [2] Best camera remote shutter releases. [www-dokumentti].  
Saatavilla: <http://www.techradar.com/news/photography-video-capture/cameras/best-camera-remote-shutter-releases-1046267/1> (Luettu 25.11.2015)
- [3] How to use a Wi-Fi-enabled DSLR with Nikon's WMU app. [www-dokumentti].  
Saatavilla: <http://www.digitalcameraworld.com/2015/07/05/how-to-use-a-wi-fi-enabled-dslr-with-nikons-wmu-app/> (Luettu 25.11.2015)
- [4] Circuitstoday.com. Basics of microcontrollers. [www-dokumentti].  
Saatavilla: <http://www.circuitstoday.com/basics-of-microcontrollers> (Luettu 24.10.2015)
- [5] What Is PICAXE? [www-dokumentti].  
Saatavilla: <http://www.picaxe.com/What-Is-PICAXE> (Luettu 2.11.2015)
- [6] TI LaunchPad. [www-dokumentti].  
Saatavilla: <http://www.ti.com/ww/en/launchpad/launchpad.html> (Luettu 2.11.2015)
- [7] Wikipedia: TI MSP430. [www-dokumentti].  
Saatavilla: [https://en.wikipedia.org/wiki/TI\\_MSP430](https://en.wikipedia.org/wiki/TI_MSP430) (Luettu 2.11.2015)
- [8] Raspberry Pi. [www-dokumentti].  
Saatavilla: <https://www.raspberrypi.org> (Luettu 2.11.2015)
- [9] Karvinen, T. ja Karvinen, K., Sulautetut, Opi rakentamaan robotteja ja muita sulautettuja järjestelmiä, 2. painos, 2010
- [10] Evans, B., Beginning Arduino Programming, 2011
- [11] Evans, M., Noble, J., Hochenbaum, J., Arduino in Action, 2013
- [12] Blum, J., Exploring Arduino: Tools and Techniques for Engineering Wizardy, 2013
- [13] Arduino. [www-dokumentti].  
Saatavilla: <http://www.arduino.cc> (Luettu 1.11.2015)
- [14] CSR BC417. [datalehti].  
Saatavilla: [www.olimex.cl/documents/msds/Datasheet%20CSR-BC417.pdf](http://www.olimex.cl/documents/msds/Datasheet%20CSR-BC417.pdf) (Luettu 2.11.2015)
- [15] Songle SRD-05VDC-SL-C. [datalehti].  
Saatavilla: <https://www.ghielectronics.com/downloads/man/20084141716341001RelayX1.pdf> (Luettu 2.11.2015)
- [16] Atmel Atmega328P. [datalehti].  
Saatavilla: <http://www.atmel.com/Images/doc8161.pdf> (Luettu 2.11.2015)

- [17] VARTA 4922. [datalehti].  
Saatavilla: <http://www.houseofbatteries.com/documents/4922.pdf> (Luettu 25.11.2015)
- [18] Assembly and Operating Instructions for HiViz.com Kits. [www-dokumentti].  
Saatavilla: [http://www.hiviz.com/kits/instructions/rsr\\_inst.htm](http://www.hiviz.com/kits/instructions/rsr_inst.htm) (Luettu 25.11.2015)
- [19] Chapter 2: Components of MCU. [www-dokumentti].  
Saatavilla: <https://www.techshopbd.com/tutorial-categories/avr-tutorial/32/chapter-2-components-of-mcu-techshop-bangladesh> (Luettu 25.11.2015)

Name	Processor	Operating Voltage/Input Voltage	CPU Speed	Analog In/Out	Digital IO/PWM	EEPROM [KB]	SRAM [KB]	Flash [KB]	USB	UART
Uno	ATmega328	5 V/7-12 V	16MHz	6/0	14/6	1	2	32	Regular	1
Due	AT91SAM3X8E	3.3 V/7-12 V	84 MHz	12/2	54/12	-	96	512	2 Micro	4
Leonardo	ATmega32u4	5 V/7-12 V	16MHz	12/0	20/7	1	2.5	32	Micro	1
Mega 2560	ATmega2560	5 V/7-12 V	16MHz	16/0	54/15	4	8	256	Regular	4
Mega ADK	ATmega2560	5 V/7-12 V	16MHz	16/0	54/15	4	8	256	Regular	4
Micro	ATmega32u4	5 V/7-12 V	16MHz	12/0	20/7	1	2.5	32	Micro	1
Mini	ATmega328	5 V/7-9 V	16MHz	8/0	14/6	1	2	32	-	-
Nano	ATmega168	5 V/7-9 V	16MHz	8/0	14/6	0.512	1	16	Mini-B	1
	ATmega328					1	2	32		
Ethernet	ATmega328	5 V/7-12 V	16MHz	6/0	14/4	1	2	32	Regular	-
Esplora	ATmega32u4	5 V/7-12 V	16MHz	-	-	1	2.5	32	Micro	-
ArduinoBT	ATmega328	5 V/2.5-12 V	16MHz	6/0	14/6	1	2	32	-	1
Fio	ATmega328P	3.3 V/3.7-7 V	8MHz	8/0	14/6	1	2	32	Mini	1
Pro (168)	ATmega168	3.3 V/3.35-12 V	8MHz	6/0	14/6	0.512	1	16	-	1
Pro (328)	ATmega328	5 V/5-12 V	16MHz	6/0	14/6	1	2	32	-	1
Pro Mini	ATmega168	3.3 V/3.35-12 V	8MHz	6/0	14/6	0.512	1	16	-	1
		5 V/5-12 V	16MHz							
LilyPad	ATmega168V	2.7-5.5 V	8MHz	6/0	14/6	0.512	1	16	-	-
	ATmega328V	V/2.7-5.5 V								
LilyPad USB	ATmega32u4	3.3 V/3.8-5V	8MHz	4/0	9/4	1	2.5	32	Micro	-
LilyPad Simple	ATmega328	2.7-5.5 V/2.7-5.5 V	8MHz	4/0	9/4	1	2	32	-	-
LilyPad SimpleSnap	ATmega328	2.7-5.5 V/2.7-5.5 V	8MHz	4/0	9/4	1	2	32	-	-
Yun	ATmega32u4	5 V	16MHz	12/0	20/7	1	2.5	32	Micro	1

```
*testi3.py - C:\Python34\testi3.py (3.4.2)*
File Edit Format Run Options Windows Help

from tkinter import *
import struct
import sys
import serial
import time

def aika():
    val = str(Entry.get(kentta))+"/"+
    tulos = "Valoitus aika: " + val
    teksti_ikkuna['text'] = tulos
    Port.write(str.encode(val))

def lopeta():
    ikkuna.destroy()
    sys.exit(0)
    Port.close()

def laukaise():
    tulos2 = "Laukaistu!"
    teksti_ikkuna2['text'] = tulos2
    Port.write(str.encode("#"))

Port = serial.Serial('COM11', 9600)
time.sleep(2)

ikkuna = Tk()
ikkuna.title("Kaukolaukaisin")
ikkuna.geometry("250x100")

ruutu1 = Frame(ikkuna, borderwidth = 3)
ruutu1.pack()
ruutu2 = Frame(ruutu1, borderwidth = 3)
ruutu2.pack(side = RIGHT)

kentta = Entry(ruutu1)
kentta.pack()

teksti_ikkuna = Label(ruutu1, text = "")
teksti_ikkuna.pack()

teksti_ikkuna2 = Label(ruutu2, text = "")
teksti_ikkuna2.pack(side = BOTTOM)

tarkenna_nappi = Button(ruutu2, text = "Tarkenna", \
                        width = 12, command = aika)
tarkenna_nappi.pack()

laukaise_nappi = Button(ruutu2, text = "Laukaise", \
                        width = 12, command = laukaise)
laukaise_nappi.pack()

lopeta_nappi = Button(ruutu2, text = "Lopeta", \
                      width = 12, command = lopeta)
lopeta_nappi.pack(side = BOTTOM)

ikkuna.mainloop()

#haetaan tarvittavat kirjastot ikkunan luomista, sarjayhteyttä
#ja aikaa varten

#määritellään komento aika
#luetaan tiedot kentästä, lisätään / perään ja tallennetaan muuttujaan val
#tallennetaan tulos-muuttujaan val-muuttujan sisältö ja lisätään teksti alkuun
#lähetetään tulos-muuttujan sisältö tekstikenttään teksti_ikkuna
#kirjoitetaan Port-muuttujaan val-muuttujan sisältö

#määritellään komento lopeta
#sammutetaan ikkuna
#poistutaan ohjelmasta
#suljetaan sarjayhteys

#määritellään komento laukaise
#tallennetaan teksti muuttajaan tulos2
#lähetetään muuttujan tulos2 tekstikenttään teksti_ikkuna2
#kirjoitetaan Port-muuttujaan merkki "#"

#annetaan muuttuja Port, joka lähettää annetun tiedon sarjayhteydellä
#tauko 2ms

#luodaan ikkuna, annetaan otsikko ja ikkunan koko

#luodaan ikkunaan 2 ruutua

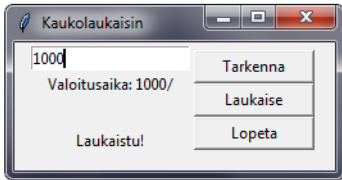
#luodaan ikkunaan syöttökenttä tekstille

#luodaan syöttökentän alle tekstikenttä

#luodaan ruudun alaosaan tekstikenttä

#luodaan toiseen ruutuun näppäimet Tarkenna, Laukaise ja Lopeta
#sekä annetaan niille koko ja suoritettava komento

#luodaan ohjelmalle loop-komento
```



Ln: 56 | Col: 60

```

int tark = 7; //pinnien nimeäminen
int lauk = 8;
unsigned long sarjadata = 0; //muuttujien alustus
int bita;
int bitb;
int lopetus;
int aika;

void setup(){
  pinMode(tark, OUTPUT); //pinnien asetus ulostuloiksi
  pinMode(lauk, OUTPUT);
  Serial.begin(9600); //sarjaliikenteen käynnistys
}

void loop(){ //loop-funktion määrittäminen
  if(Serial.available()){ //jos sarjaliikennettä havaitaan, jatketaan, muuten alkuun
    haedata(); //kutsu aliohjelmaa haedata
    digitalWrite(tark, HIGH); //tarkennuspinni päälle eli rele kiinni
    laukaise(); //kutsu aliohjelmaa laukaise
  }
}

long haedata(){ //haedata-aliohjelma
  while (bita != '/') { //aliohjelman luoppaus, kunnes merkki / havaitaan
    bita = Serial.read(); //seuraavilla 4:llä rivillä luodaan kokonaisluku merkkijonosta,
    Serial.println(bita); //jossa katkaisijana toimii /-merkki
    if (bita > 0 && bita != '/') {
      sarjadata = sarjadata * 10 + bita - '0';
    }
  }
  Serial.println(sarjadata); //saatu kokonaisluku kirjoitetaan sarjaliikenneporttiin
  aika = sarjadata; //luku tallennetaan aika-muuttujaan
  return sarjadata; //palautetaan sarjadata-muuttuja
  bita = 0; //nollataan bita
}

long laukaistaanko(){ //laukaistaanko-aliohjelma
  while (bitb != '#') { //aliohjelman luoppaus, kunnes merkki # havaitaan
    bitb = Serial.read(); //aliohjelmassa luetaan sarjaliikennettä ja tallennetaan
    Serial.println(bitb); //muuttujaan bitb
  }
  bitb = 0; //nollataan bitb
}

void laukaise(){ //laukaise-aliohjelma
  laukaistaanko(); //kutsutaan aliohjelmaa laukaistaanko
  digitalWrite(lauk, HIGH); //laukaisupinni päälle eli rele kiinni
  delay(aika); //viive eli valoitusaika
  digitalWrite(lauk, LOW); //vapautetaan laukaisu- ja tarkennusreleet
  digitalWrite(tark, LOW);
  nollaa(); //kutsutaan aliohjelmaa nollaa, jossa alustetaan muuttujat nolllaksi
}

```